Intel® IXP400 Software: VLAN and QoS Application Version 2.2

Release Notes

August 2, 2007

Document Number: 301925-005

Intel may make changes to specifications and product descriptions at any time, without notice.

The Intel IXP400 Software may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request.

MPEG is an international standard for video compression/decompression promoted by ISO. Implementations of MPEG CODECs, or MPEG enabled platforms may require licenses from various entities, including Intel Corporation.

Contact your local Intel sales office or your distributor to obtain the latest specifications and before placing your product order.

Copies of documents which have an order number and are referenced in this document, or other Intel literature may be obtained by calling 1-800-548-4725 or by visiting Intel's website at http://www.intel.com.

Contents

1.0 Introduction

The Intel® IXP400 Software VLAN and QoS Application v2.2, which runs on the Intel® IXDP425 Development Platform, Intel® IXDP465 Development Platform and the Intel® IXP435 Multi-Service Residential Gateway Reference Platform, demonstrates IEEE 802.1Q VLAN and IEEE 802.1p User Priority to Traffic Class Mapping functionality. The application relies on Intel® IXP400 Software and uses the Ethernet bridging software and NPE Ethernet device drivers for the supported operating systems.

This document provides important information for using the Intel® IXP400 Software VLAN and QoS Application v2.2. This release notes provides instructions for obtaining, installing, compiling and using the user interface of the provided code. For detailed information on the architecture and APIs provided with the VLAN and QoS Example Code, refer to the *Intel ® IXP400 Software VLAN and QoS Application v2.0 Programmer's Guide*.

2.0 What's New

The Intel® IXP400 Software VLAN and QoS Application v2.2 is enabled with Linux* kernel 2.6.16.11 distributed by Timesys*.

3.0 Additional Resources

The following table lists the additional documents that are relevant to this release. These documents are available either through your field representative or at the following web site:

http://www.intel.com/design/network/products/npfamily/docs/ixp4xx.htm

Title	Document Number
Intel ® IXP400 Software VLAN and QoS Application v2.0 Programmer's Guide	301925
Intel® IXP400 Software Version 2.4 Release Notes	N/A†
Intel® IXP400 Software Programmer's Guide	252539
Intel® IXP400 Software Specification Update	273795
Intel® IXP400 Software: RedHat Boot loader v.2.04 Software - Release Notes*	N/A†

† This document is on the IXP400 software Web page at: http://www.intel.com/design/network/products/npfamily/ixp400_archives.htm

4.0 Obtaining the Intel® IXP400 Software, VLAN and QoS Example Code

The VLAN and QoS Example Code are provided as patches and additional source code that build upon the Intel® IXP400 Software base. The IXP400 software has certain prerequisites, and the VLAN and QoS Example Code have some additional requirements.

The following are the prerequisites for a target platform running on Linux* kernel 2.6.16.11.

- **Intel® IXP400 Software v2.4:** To obtain this software, go to the following Web site:
  http://www.intel.com/design/network/products/npfamily/ixp400_archives.htm
- **Intel® IXP400 Software Linux Ethernet Device Driver v1.7:** These files are available from:
  http://www.intel.com/design/network/products/npfamily/ixp400_osc.htm
- **RedBoot* boot-loader:** The Intel® IXDP425 Development Platform, Intel® IXDP465 Development Platform and Intel® IXP435 Multi-Service Residential Gateway Reference Platform come with a suitable version of RedBoot in flash. For your convenience, the source code and tool chain are available from:
  http://www.intel.com/design/network/products/npfamily/ixp400_archives.htm
- **Intel® IXP400 Software VLAN and QoS Application v2.2:** This software comes in two parts for Linux. **Part 1** is a patch to the Intel® IXP400 Software Linux Device Driver, version 1.7. **Part 2** is the VLAN and QoS Example Code source for Linux. Both parts are required.

The VLAN and QoS Example Code is provided in source code, and contains patches to some existing Intel and third-party content. For more information on the architecture and APIs provided with the VLAN and QoS Example Code, refer to the *Intel ® IXP400 Software VLAN and QoS Application v2.0 Programmer's Guide*. The application is contained in two separate downloads.

- **BSD_ixp400SwVlanQosApp-2_2.zip**

This file contains the various application source code components for Linux* kernel 2.6.16.11. The .zip file package content is shown below:

```
ixp400_xscale_sw\src\ixVlanQos\* (--source files for VLAN and QoS module)

ixp400_xscale_sw\src\vqconfig\* (--source files for VLAN QoS configuration
application)
```

- **ixp400SwVlanQos-2_2_EDD-1_7.patch.zip**

This file contains a single patch file for the Intel® IXP400 Software Ethernet Device Driver, version 1.7. It adds the VLAN and QoS functionality to the Ethernet driver.

5.0 Installing and Building the Software — Linux* Kernel 2.6.16.11

5.1 Installing the Linux* Kernel 2.6.16.11 and LSP

1. Ensure the cross-development environment is configured properly, including connecting the IXDP425 development platform/IXDP465 development platform/Intel IXP435 reference platform to the host development system.
2. Ensure that the kernel 2.6.16.11 and required LSP is installed properly according to the instructions provided in the *Intel® IXP400 Software Version 2.4 Release Notes*.

3. Create a working directory and copy the LSP kernel source tree into the working directory by typing the following command:

```
$ mkdir <work_dir>

$ cp –a <PATH_TO_LSP>/kernel-2.6.16.11-ts.ixp435.1 ./
```

This places the LSP kernel in a directory under `<work_dir>` as `kernel-2.6.16.11-ts.ixp435.1.`The directory can be renamed as `linux` for convenience. Alternatively, create a soft link named `linux` that points to the directory, so that the LSP kernel version is maintained in the directory name. The path to the LSP will be noted as `<linux>` from this point onwards.

5.2 Installing IXP400 Software Access Library v2.4, NPE Ethernet Device Driver and VLAN & QoS Common Module

1. Install the Intel® IXP400 Software v2.4 by following the instructions below:

   Note: The following steps use directory path naming conventions documented in the *Intel® IXP400 Software Version 2.4 Release Notes*.

   - Obtain the IXP400 software Access Library v2.4 and unzip the two zip files listed as below. The source trees of Access Library must be installed in the working directory.

```
$ cd <work_dir>

$ unzip <path_to_file>/BSD_ixp400AccessLibrary-2_4.zip

$ unzip <path_to_file>/IPL_ixp400NpeLibrary-2_4.zip
```

   - At this point, the working directory has the following structure:

```
<work_dir>

  | -- ixp400_xscale_sw

  | -- ixp_osal

 | -- linux
```

2. Install the Intel® IXP400 Software Linux Ethernet Driver.

   - Unzip the `GPL_ixp400LinuxEthernetDriverPatch-1_7.zip`, which contains a patch file for the Ethernet device driver. Use the following commands to unzip the zip file, apply the patch and install the Ethernet device driver:

```
$ unzip -d <temp_dir> <path_to_file>/GPL_ixp400LinuxEthernetDriverPatch-1_7.zip
  inflating: temp/ixp400_eth.c
  inflating: temp/ixp400LinuxEthernetDriver-1_7.patch
  inflating: temp/Readme.txt
$ cd <work_dir>/linux
$ patch -p1 < <temp_dir>/ixp400LinuxEthernetDriver-1_7.patch
$ cp <temp_dir>/ixp400_eth.c ./drivers/net/
```

3. Install the common module and configuration utility (source code) of the Intel® IXP400 Software VLan and QoS v2.2

   - Unzip the BSD_ixp400SwVlanQosApp-2_2.zip into <work_dir> (the working directory must be the same as the one where Intel® IXP400 Software is installed) using the following instructions. Two new components (result of the installation) ixVlanQos and vqconfig can be found under the <work_dir>/ixp400_xscale_sw/src/:

```
$ unzip -d <work_dir> <path_to_file>/BSD_ixp400SwVlanQosApp-2_2.zip
```

   - Edit the environment variables accordingly in either <work_dir>/ixp400_xscale_sw/buildUtils/environment.linux.sh or <work_dir>/ixp400_xscale_sw/buildUtils/environment.linux.csh depending on which shell is used.
     Please refer to *Intel® IXP400 Software Version 2.4 Release Notes* for further information on the environment variables. Add the following line also to the end of file:

```
export IX_INCLUDE_VLAN_QOS=1
```

   - Set up the environment variables using one of the following commands.

```
$ source ./ixp400_xscale_sw/buildUtils/environment.linux.sh
```

   or

```
$ source ixp400_xscale_sw/buildUtils/environment.linux.csh
```

   - Edit the makefile and add the lines (according to platform) to the place as show in red typeface below:

   — For the Intel® IXP435 multi-service residential gateway reference platform: <work_dir>/ixp400_xscale_sw/Makefile_ixp43x

   — For the Intel® IXDP425 development platform: <work_dir>/ixp400_xscale_sw/Makefile_ixp42x

   — For the Intel® IXDP465 development platform: <work_dir>/ixp400_xscale_sw/Makefile_ixp46x

```
$ BI_ENDIAN_COMPONENTS := atmdAcc atmm atmsch …

ifdef IX_INCLUDE_VLAN_QOS
BI_ENDIAN_COMPONENTS += ixVlanQos
endif
```

4. Apply a patch for the Ethernet Device Driver.

   The Ethernet device driver must be patched for integrating VLAN and QoS functionality. The patch file is `ixp400SwVlanQos-2_2_EDD-1_7.patch` and should be applied to the LSP source tree. Use the following commands to apply this patch:

```
$ unzip –d <temp_dir> <path_to_file>/ixp400SwVlanQos-2_2_EDD-1_7.zip
extracting: ixp400SwVlanQos-2_2_EDD-1_7.patch
$ cd <work_dir>/linux
$ patch -p1 < <temp_dir>/ixp400SwVlanQos-2_2_EDD-1_7.patch
```

Note: You should have installed Access Library source files before applying this patch.

5.3 Building the VLAN and QoS Example Code

5.3.1 Compiling the Linux Kernel Image

1. Ensure the environment is configured and set.
2. Enter to the Linux source tree and Linux* kernel configuration menu by using the following commands:

```
For the IXP435 reference platform
$ cd linux
$ make $IX_KBUILD_CROSSCOMPILE kixrp435_defconfig
$ make $IX_KBUILD_CROSSCOMPILE menuconfig
```

```
For the IXDP425 development platform
$ cd linux
$ make $IX_KBUILD_CROSSCOMPILE ixp42x_defconfig
$ make $IX_KBUILD_CROSSCOMPILE menuconfig
```

```
For the IXDP465 development platform

$ cd linux

$ make $IX_KBUILD_CROSSCOMPILE ixp46x_defconfig

$ make $IX_KBUILD_CROSSCOMPILE menuconfig
```

3. In Linux* kernel configuration menu, select the following items, and type "M" in item `<Intel IXP400 Ethernet Device support>`

```
Device Drivers --> Network device support --> Ethernet (10 or 100Mbit) -->

<M> Intel IXP400 Ethernet Device support
```

Note: The `Intel IXP400 Ethernet Device Driver NAPI support` option (one of the several sub-options under the Intel IXP400 Ethernet Device support) should NOT be enabled. The enabling of NAPI mechanism drops the performance of QoS traffic controlling function.

Modify other Linux configurations as needed.

To use the Ethernet bridging function to perform VLAN switching, enable:

```
Networking --> Networking options -->

<*> 802.1d Ethernet Bridging
```

4. Save and exit from the menu configuration.
5. Build the Linux kernel image.

```
$ make $IX_KBUILD_CROSSCOMPILE zImage
```

Copy the `zImage` from `<workdir>/linux/arch/arm/boot/` to the `tftpboot` directory, typically `/tftpboot`. This can be booted using the RedBoot bootloader.

5.3.2 Compiling Loadable Kernel Modules

1. To build the configured kernel modules, including the NPE Ethernet driver (`ixp400_eth.ko`), use the following command:

```
$ cd linux

$ make $IX_KBUILD_CROSSCOMPILE modules
```

2. Copy the loadable modules, for example the `ixp400_eth.ko`, in `<work_dir>/linux/drivers/net` into the embedded target file system on the host machine <TARGET_FILE_SYSTEM>/, for example:

```
$ cp <workdir>/linux/drivers/net/ixp400_eth.ko /opt/timesys/
linux/6.1/armv5b-std/rfs/
```

5.3.3 Compiling the Access Library and the Common Module of VLAN and QoS

1. Ensure the environment is configured and set.
2. Use the following commands to compile the Access Library:

```
$ cd <work_dir>/ixp400_xscale_sw/

$ make ixp400
```

3. The resulting modules and NPE image are located in one of the following directories below, as appropriate:
   - For the IXP435 reference platform:

     `ixp400_xscale_sw/ixp43x/lib/linuxbe/`
   - For the IXDP425 development platform:

     `ixp400_xscale_sw/lib/linuxbe`
   - For the IXDP465 development platform:

     `ixp400_xscale_sw/ixp46x/lib/linuxbe`

   Place the required files (in this case, `ixp400.ko` and `IxNpeMicrocode.dat`) into the target file system directory.

5.3.4 Compiling the VLAN & QoS Configuration Utility

1. To build the configuration utility, use the following commands:

```
$ cd <work_dir>/ixp400_xscale_sw/src/vqconfig

$ make -f component.mk
```

2. Install the configuration utility by copying the utility to target file system directory using the following command:

```
  $ cp vqconfig /opt/timesys/linux/6.1/armv5b-std/rfs/
```

5.4 Running the VLAN and QoS Example Code

1. At this point, boot the IXP435 reference platform/ IXDP465 development platform/IXDP425 development platform using RedBoot and the `zImage` created above. Refer to the *Intel® IXP400 Software Version 2.4 Release Notes* for instructions on this process.
2. When Linux* has been booted on the IXP435 reference platform/ IXDP465 development platform/IXDP425 development platform, the following 3 module files have to be added to Linux* kernel in the order: `ixp400.ko` first, next `IxNpeMicrocode.dat`, and finally `ixp400_eth.ko`. After successfully adding them, an Ethernet driver supporting the VLAN and QoS feature is set up and enabled for the NPE Ethernet ports. Use `ifconfig` to configure the network interfaces exported by NPE, that is ixp0, ixp1, and ixp2 (if exists). To configure the VLAN and QoS feature to the NPE Ethernet

ports, use vqconfig. Starting vqconfig without any argument will list a brief help menu for usage.

```
For the IXP435 reference platform

login : root

$ mknod /dev/ixNpe c 241 0

$ insmod ixp400.ko

$ cat IxNpeMicrocode.dat > /dev/ixNpe

$ insmod ixp400_eth.ko no_phy_scan=1

$ ifconfig ixp1 [IP address B]

$ ifconfig ixp2 [IP address A]

$ ./vqconfig
```

```
For the IXDP465 development platform or the IXDP425 development platform

login : root

$ mknod /dev/ixNpe c 241 0

$ insmod ixp400.ko

$ cat IxNpeMicrocode.dat > /dev/ixNpe

$ insmod ixp400_eth.ko

$ ifconfig ixp0 [IP address B]

$ ifconfig ixp1 [IP address C]

$ ./vqconfig
```

Note: ixp0 = NPE-B (not applicable for the IXP435 reference platform)

ixp1 = NPE-C

ixp2 = NPE-A (not applicable for IXDP425 development platform)

6.0 Structure of Target Files

The executable image of the software is generated and separated into three discrete components. The three components are common module, NPE Ethernet driver, and configuration utility. The following sections contain brief explanations of these components.

6.1 Common Module

The common module is the core component of the VLAN and QoS Application and it contains all of the major functionalities. This module is independent to the operating system it is running on. Its executable image is included in the target file (object file) of the IXP400 Access Library. Linking and running the Access Library brings up this common module.

6.2 NPE Ethernet Driver

This software contains the modifications to existing NPE Ethernet drivers in order to invoke the VLAN and QoS functionalities via the management interfaces by the common module. Refer to the source files of the modified Ethernet driver to understand the usage of VLAN and QoS APIs. Patches are available for the following Ethernet driver:

- IXP400 NPE Ethernet driver v1.7 for Linux* kernel 2.6.16.11

The modified portion of the Ethernet driver is included in the driver itself. No additional target files are generated.

Note: The NAPI feature is not recommended to be used in conjunction with VLAN QoS software to avoid undesired system behavior.

6.3 Configuration Utility

The VLAN and QoS configuration utility allows the user to communicate with the VLAN and QoS application. The user enters command lines to the utility. The format of the commands is shown in Section 6.4. The utility accepts the command lines and translates them into appropriate IOCTL actions. The utility then submits the IOCTL actions to the NPE Ethernet driver and receives responses from the driver or the system. The utility translates the received responses into human-readable format and displays them on-console.

On Linux systems, the utility is built as an executable program file running in user mode. The filename is `vqconfig`.

6.4 Launching the Configuration Utility

Launch the configuration utility by typing `vqconfig` at the shell prompt. The configuration usage is displayed and then enters an interactive state waiting for input commands from the user. In this state a “>” is displayed at the start of the input line. Type `quit` to exit the configuration utility.

```
#./vqconfig

Usage is displayed..........

>

>

> quit

Program quited.

#
```

Under Linux, the command lines can be added to the vqconfig invocation. In this case, the utility would execute the commands, display the results, and exit.

```
# ./vqconfig set ixp0 vlan=on

Results displayed..........

#
```

6.5 Usage Commands

The following sections give a brief overview of the functionality of the configuration utility. Description, format, parameters and examples are given for each command.

6.5.1 add – MAC Address or IP Rule

Description:

- Add a MAC address for MAC-based VLAN classification
- Add an IP or IPv6 rule for protocol-based VLAN classification

Format:

```
add {IP-RULE|IP6-RULE|MAC-RULE} TAG
```

Parameters:

`IP-RULE`: Describing an IP rule for protocol-based VLAN classification associated to a VLAN group

`IP6-RULE`: Describing an IPv6 rule for protocol-based VLAN classification associated to a VLAN group

`MAC-RULE`: Describing a MAC address for MAC-based VLAN classification associated to a VLAN group

`TAG`: the VLAN tag which describing a VLAN group. It contains the VLAN ID and user priority

Examples:

```
# ./vqconfig

> add ip sip=192.168.80.0/24 vid=2

> add ip sip=192.168.80.0/24 dip=192.168.100.252/30 vid=3 priority=5

> add ip tcp vid=3 priority=6

> add ip udp dport=18 sport=20 vid=4



> add ipv6 sip=00AA:BBCC::0102:0303 ah spi=123 vid=5

> add ipv6 sip=00AA:BBCC::0102:0505 tcp sport=21 vid=5



> add mac=0a:bc:de:01:01:01 vid=6

> add mac=0a:bc:de:02:02:02 vid=7 priority=7
```

6.5.2 del – MAC Address or IP Rule

Description:

- Delete a MAC address from MAC-Based VLAN classification
- Delete an IP or IPv6 rule from Protocol-Based VLAN classification

Format:

```
del { IP-RULE | IP6-RULE | MAC-RULE | rule=RULE-ID }
```

Parameters:

IP-RULE: Describing an IP rule for protocol-based VLAN classification associated to a VLAN group

IP6-RULE: Describing an IPv6 rule for protocol-based VLAN classification associated to a VLAN group

MAC-RULE: Describing a MAC address for MAC-based VLAN classification associated to a VLAN group

RULE-ID: Unique number for each IP/IPv6 rule or MAC address

Examples:

```
# ./vqconfig

> del ip sip=192.168.80.0/24

> del ip sip=192.168.100.0/24 dip=192.168.100.252/30
```

6.5.3 join/leave – NPE Ethernet Port

Description:

- Join/leave a NPE Ethernet port to/from a VLAN group

Format:

```
join INTERFACE vid=VID
leave INTERFACE vid=VID
```

Parameters:

`INTERFACE`: Network Interface of NPE Ethernet port

`VID`: VLAN Identifier (1~4094)

Examples:

```
# ./vqconfig

> join ixp0 vid=3

> leave ixp1 vid=2
```

6.5.4 set – VLAN Tag

Description:

- Set the default VLAN tag for a NPE Ethernet port

Format:

```
set INTERFACE tag TAG
```

Parameters:

`INTERFACE`: Network Interface of NPE Ethernet port

`TAG`: the VLAN tag which describing a VLAN group. It contains the VLAN ID and user priority

Examples:

```
# ./vqconfig

> set ixp0 tag vid=2

> set ixp1 tag vid=3 priority=1
```

6.5.5 set – Acceptable Frame Type

Description:

- Set the acceptable frame type for a NPE Ethernet port

Format:

```
set INTERFACE accept=ACCEPTABLE
```

Parameters:

INTERFACE: Network Interface of NPE Ethernet port

ACCEPTABLE: Acceptable type of the ingress frame

Examples:

```
# ./vqconfig

> set ixp0 accept=all

> set ixp1 accept=tag
```

6.5.6 set – Egress Frame Type

Description:

- Set the egress frame type to a VLAN group associated with a NPE Ethernet port

Format:

```
set INTERFACE vid=VID egress=FRAME-TYPE
```

Parameters:

INTERFACE: Network Interface of NPE Ethernet port

VID: VLAN Identifier (1~4094)

FRAME-TYPE: Frame type of the egress frame

Examples:

```
# ./vqconfig

> set ixp0 vid=2 egress=tag

> set ixp0 vid=3 egress=untag
```

6.5.7 set – VLAN Membership Filtering

Description:

- Enable/disable the VLAN Membership Filtering for a NPE Ethernet port

Format:

```
set INTERFACE DIRECTION filter=STATE
```

Parameters:

INTERFACE: Network Interface of NPE Ethernet port

DIRECTION: Direction of the transporting frame (ingress or egress)

STATE: To enable or to disable

Examples:

```
# ./vqconfig

> set ixp0 ingress filter=on

> set ixp1 egress filter=off
```

6.5.8 set – VLAN Capability

Description:

- Enable/disable the VLAN capability for a NPE Ethernet port

Format:

```
set INTERFACE vlan=STATE
```

Parameters:

INTERFACE: Network Interface of NPE Ethernet port
STATE: To enable or to disable

Examples:

```
# ./vqconfig

> set ixp0 vlan=off

> set ixp1 vlan=on
```

6.5.9 set – VLAN Classifier

Description:

- Enable/disable the Protocol-Based VLAN classifier of a NPE Ethernet port
- Enable/disable the MAC-Based VLAN classifier of a NPE Ethernet port

Format:

```
set INTERFACE classifier {mac=STATE | protocol=STATE}
```

Parameters:

INTERFACE: Network Interface of NPE Ethernet port
STATE: To enable or to disable

Examples:

```
# ./vqconfig

> set ixp0 classifier mac=on

> set ixp0 classifier mac=off

> set ixp1 classifier mac=on
```

6.5.10 set – Average Rate or Ceil Rate

Description:

- Set the Average Rate of Traffic Shaper
- Set the Ceil Rate of Traffic Shaper

Format:

```
set INTERFACE DIRECTION tc=TC rate [frame=RATE] [data=RATE]
set INTERFACE DIRECTION tc=TC ceil [frame=RATE] [data=RATE]
```

Parameters:

INTERFACE: Network Interface of NPE Ethernet port
DIRECTION: Direction of the transporting frame (ingress or egress)
TC: Traffic class (0, 1, 2, 3)
RATE: The traffic rate. The unit is frame per second or bit per second

Examples:

```
# ./vqconfig

> set ixp0 ingress tc=0 rate frame=3000 data=30000

> set ixp0 ingress tc=0 ceil frame=5000 data=50000
```

6.5.11 set – Burst Size of Traffic Shaper

Description:

- Set the burst size of a Traffic Shaper

Format:

```
set INTERFACE DIRECTION tc=TC burst [frame=SIZE] [data=SIZE]
```

Parameters:

INTERFACE: Network Interface of NPE Ethernet port
DIRECTION: Direction of the transporting frame (ingress or egress)
TC: Traffic class (0, 1, 2, 3)
SIZE: The burst size in the unit of frame or data bit

Examples:

```
# ./vqconfig

> set ixp0 ingress tc=0 burst frame=30000

> set ixp0 ingress tc=1 burst data=1000000

> set ixp0 ingress tc=2 burst frame=30000 data=1000000
```

6.5.12 set – Frame Shaper

Description:

- Enable/disable a Traffic Shaper

Format:

```
set INTERFACE DIRECTION tc=TC shaper=STATE
```

Parameters:

INTERFACE: Network Interface of NPE Ethernet port
DIRECTION: Direction of the transporting frame (ingress or egress)
TC: Traffic class (0, 1, 2, 3)
STATE: To enable or to disable

Examples:

```
# ./vqconfig

> set ixp0 ingress tc=0 shaper=on

> set ixp1 egress tc=1 shaper=off
```

6.5.13 set – Mapped Traffic Class

Description:

- Set the Mapped Traffic Class for a User Priority

Format:

```
set INTERFACE DIRECTION map prio=PRIORITY tc=TC
```

Parameters:

INTERFACE: Network Interface of NPE Ethernet port
DIRECTION: Direction of the transporting frame (ingress or egress)
PRIORITY: User priority (0~7)
TC: Traffic class (0, 1, 2, 3)

Examples:

```
# ./vqconfig

> set ixp0 ingress map prio=0 tc=1

> set ixp1 egress map prio=7 tc=3
```

6.5.14 set – Mapping Tables

Description:

- Set the Mapping Table for Eight User Priorities

Format:

```
set INTERFACE DIRECTION map table=TC-TABLE
```

Parameters:

INTERFACE: Network Interface of NPE Ethernet port
DIRECTION: Direction of the transporting frame (ingress or egress)
TC-TABLE: Traffic class table

Examples:

```
# ./vqconfig
> set ixp0 ingress map table=0,0,0,0,0,1,2,3
> set ixp1 egress map table=0,0,1,1,2,2,3,3
```

6.5.15 show – MAC Addresses or IP/IPv6 Protocol Rules

- Show all MAC addresses or IP/IPv6 protocol rules

Format:

```
show [ mac | proto ] rules
```

Parameters:

None

Examples:

```
# ./vqconfig
> show mac rules
> show proto rules
```

6.5.16 show – VLAN Capabilities or Membership Table

Description:

- Show the status of VLAN capabilities of a NPE Ethernet port
- Show the VLAN membership table of a NPE Ethernet port

Format:

```
show INTERFACE { state | membership }
```

Parameters:

INTERFACE: Network Interface of NPE Ethernet port

Examples:

```
# ./vqconfig
> show ixp0 stat
> show ixp1 membership
```

6.5.17 show – Shaper Status or Mapping Table

Description:

- Show the status of shaper(s)
- Show the mapping table of traffic classes

Format:

```
show INTERFACE DIRECTION { [tc=TC] shaper | map }
```

Parameters:

`INTERFACE`: Network Interface of NPE Ethernet port
`DIRECTION`: Direction of the transporting frame (ingress or egress)
`TC`: Traffic class (0, 1, 2, 3)

Examples:

```
# ./vqconfig

> show ixp0 ingress tc=3 shaper

> show ixp0 ingress shaper

> show ixp1 egress map
```

§ §